A recurrent neural network with ever changing synapses

M Heerema [+] **and WA van Leeuwen** [*]

Institute for Theoretical Physics, University of Amsterdam, Valckenierstraat 65, 1018 XE Amsterdam, The Netherlands

Abstract. A recurrent neural network with noisy input is studied analytically, on the basis of a Discrete Time Master Equation. The latter is derived from a biologically realizable learning rule for the weights of the connections. In a numerical study it is found that the fixed points of the dynamics of the net are time dependent, implying that the representation in the brain of a fixed piece of information (e.g., a word to be recognized) is not fixed in time.

Keywords: brain, recurrent neural network, biological neural network, local learning rule, local Hebb rule, energy saving learning rule, noisy patterns, double dynamics, adaptable synapses, weights, Master Equation

1. Introduction

It is our purpose to construct and describe a neural net that can learn and retrieve patterns in a way that is biologically realizable. In an actual situation, the input to a network is noisy: the brain is confronted with patterns that are similar, but not identical. Therefore, we are going to model the training phase of a neural network by considering p sets, Ω^μ, of similar patterns $\boldsymbol{x}$, centered around p typical patterns, $\boldsymbol{\xi}^\mu$ ($\mu = 1, \ldots, p$).

In the existing literature, learning rules are used which are based on typical patterns $\boldsymbol{\xi}^\mu$, and not on sets of similar patterns Ω^μ. This is biologically unrealistic: a child does not learn 'standard words', $\boldsymbol{\xi}^\mu$, pronounced by a 'standard speaker', but hears the same word pronounced by different speakers in different ways, i.e., the child is exposed to sets Ω^μ. In order to model biologically realistic learning, we use a learning rule which contains patterns $\boldsymbol{x}$ belonging to learning sets Ω^μ rather than the patterns $\boldsymbol{\xi}^\mu$ alone.

We show that when this learning rule, based on noisy input patterns is used, the network evolves to values for the strengths of the synaptic connections, usually called 'weights', that fluctuate with respect to certain fixed asymptotic values. For an actual brain this corresponds to the fact that the confrontation with input data leads to synaptic connections that change in strength, through all of their lifetime, but in such a way that there is stability in what it stores and recollects. When a biological neural network gets as input a pattern that it has learned a long time ago

[+] E-mail address: heerema@wins.uva.nl

[*] E-mail address: leeuwen@wins.uva.nl

already, and which, via the tuning of the synapses, has found a firm and fixed place on the substrate formed by the neural tissue, it nevertheless changes the synapses. This is not necessary of course, but can not be circumvented in an actual biological network. Input always changes the connections, since there is no way for an individual neuron to know whether or not a pattern has been encountered earlier. All this means that the learning rule must be such that the learning speed is not too large [1]: new information might otherwise destroy too much of the old information and, hence, the network's functioning.

So far, we have been considering the situation that patterns are presented to the network and that, through some learning rule, synaptic connections are changed [8]. A next point to discuss is how an actual, biological, neural net, while changing its connections continuously, can nevertheless recognize a pattern. In other words, what is the difference, for a neural net, between an old, i.e., already stored pattern, and a new, unknown pattern? As noted before, also an old pattern is 'stored', in the sense that it gives rise to a change of the connections. The answer to the question is that albeit that each individual neuron and each synapse reacts independently of the fact whether a pattern has been stored or not, the net as a whole reacts differently in these two cases: see section 3, in particular figure 4.

Neural nets have been studied with fixed [2, 3], and with adapting —or dynamic— synapses [1, 4, 5]. The latter neural nets are also called nets with double dynamics [6]. In the context of spin glasses one speaks of coupled dynamics [7]. In the last part of this article, we will be concerned with neural networks with double dynamics: see section 3.5.

We study what happens in a net with ever changing connections, by comparing what happens when a pattern is presented to the net that has been learned before to what happens when this is not the case. These two cases are investigated numerically on the basis of a particular learning rule, for which we have chosen the one we derived earlier [8]: it is a Mixed Hebbian-Anti-Hebbian, Hopfield like, learning rule, which is non-symmetric with respect to post- and pre-synaptic input, and which contains, moreover, a post-synaptic potential dependent factor. We found this rule assuming that building and destroying of a synapse costs biochemical energy, and by requiring, at the same time, that the energy needed to change a neural network be minimal. We suppose that the patterns $\boldsymbol{x} \in \Omega^\mu$ that are presented to the net (the various ways in which one and the same 'word' is presented) are chosen randomly from a set of patterns distributed around a set of p typical patterns $\boldsymbol{\xi}^\mu$ (the p 'standard words' to be learned).

Random processes can often be described in a useful way via a so-called Master Equation for the relevant random variables [9, 10]. We therefore start, in section 2, by deriving a Discrete Time Master Equation for the random variables in question, namely the weights w_{ij} of the connections of the network. Usually, a Master Equation is solved going from discrete time to continuous time, which always entails some essential difficulties [11, 12, 13]. Such a transition to a process that is continuous in time is often advantageous, since a differential equation, in general, is easier to solve than a difference equation. In our approach, the transition to the differential equation could be circumvented, since we had in this case at our disposal a tool that turned out to enable us to directly solve the difference equation itself: the Gauss-Seidel iterative method.

A question one might raise is whether a system with ever changing connections will ever achieve some kind of stationary state. A numerical study can not easily

answer this question, since the fluctuations of the weights are quite wild (see figure 2). We therefore performed an analytic study, based on the particular learning rule used throughout our work. We found that the system's weights will fluctuate around certain asymptotic values, and that the last stored pattern that has given rise to a fixed point is roaming over Ω^μ, the collection of patterns around a typical pattern $\boldsymbol{\xi}^\mu$. All this can be rephrased by stating that both the neural net itself and its particular states, the fixed points, wiggle around average values: the ever changing mind is, in some sense, stable.

The above can be summarized as follows. In section 2 we derive the Discrete Time Master Equation for the weights of a neural network, from a learning rule, and solve this equation analytically. In section 3 we have two objectives: firstly, to check numerically the analytical result of section 2, and, secondly, to study the implications of double dynamics.

2. The weights of a network trained with noisy patterns

In a preceding article we derived what we have called an 'energy saving learning rule'. When at time t_n $(n = 0, 1, \ldots)$ the weights are given by $w_{ij}(t_n)$, and if, thereupon, a pattern $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)$ is presented to a net of N neurons, then the weights are changed according to the rule

$$w_{ij}(t_{n+1}) = w_{ij}(t_n) + \Delta w_{ij}(t_n) \qquad (j \in V_i) \tag{1}$$

where

$$\Delta w_{ij}(t_n) = \eta_i[\kappa - \gamma_i(\boldsymbol{\xi}, \boldsymbol{w}_i(t_n))](2\xi_i - 1)\xi_j \qquad (j \in V_i) \tag{2}$$

(see [8], equations (41) and (42)). The index n in t_n labels subsequent moments of the net: t_0 is the initial time, where the weights have there initial values, $w_{ij}(t_0)$. In these equations V_i is the collection of indices j with which neuron i is connected via adaptable, non-zero synapses. Furthermore, for the so-called stability coefficients we used the abbreviation

$$\gamma_i(\boldsymbol{x}, \boldsymbol{w}_i(t)) := (\sum_{l=1}^{N} w_{il}(t)x_l - \theta_i)(2x_i - 1) \qquad (i = 1, \ldots, N) \tag{3}$$

where $\boldsymbol{w}_i := (w_{i1}, \ldots, w_{iN})$. The quantity arises naturally in case the dynamics of the network is taken to be given by equation (51) below; see, e.g., [8]. In the learning rule (3) occur two quantities, η_i and κ. For a non-biological system they can be expressed in terms of properties of the neural net and as a function of the patterns $\boldsymbol{\xi}^\mu$ to be stored in the net: for η_i, the so-called learning rate, see equations (42) and (65) of [8]; for κ, the so-called margin parameter, see [14]. For biological systems, the coefficients η_i and κ are replaced by suitable constants: for η_i see [8], section 6; for κ see [14], section 3.

It was shown that a repeated application $(n \to \infty)$ of the rule (2) gave rise to the following expression for the weights at some —finite or infinite— time t_∞

$$w_{ij}(t_\infty) = \begin{cases} w_{ij}(t_0) + N^{-1} \displaystyle\sum_{\mu,\nu=1}^{p} [\kappa - \gamma_i^\mu(t_0)](2\xi_i^\mu - 1)(C_i^{-1})^{\mu\nu}\xi_j^\nu & (j \in V_i) \\ w_{ij}(t_0) & (j \in V_i^c) \end{cases} \tag{4}$$

where $\gamma_i^\mu(t_0) := \gamma_i(\boldsymbol{\xi}^\mu, \boldsymbol{w}_i(t_0))$ and where $C_i^{\mu\nu}$ is the reduced correlation matrix, defined by

$$C_i^{\mu\nu} := N^{-1} \sum_{k \in V_i} \xi_k^\mu \xi_k^\nu \qquad (5)$$

(see [8] equation (52)). This type of learning, and the ensuing expression for the weights, $w_{ij}(t_\infty)$, correspond to the idealized situation of ideal, i.e., unperturbed, input.

2.1. The energy saving learning rule for learning with noise

In a realistic situation, however, the repeated training of a network will not take place with patterns $\boldsymbol{\xi}^\mu$ which remain exactly the same throughout all of the training process. Each time a certain pattern $\boldsymbol{\xi}^\mu$ is presented to the network, it may be slightly different. Therefore, rather than studying what happens when patterns $\boldsymbol{\xi}^\mu$ are presented, we will study what happens when patterns $\boldsymbol{x}$ are learned, which belong to sets of patterns Ω^μ clustered around typical patterns $\boldsymbol{\xi}^\mu$ ($\mu = 1, \ldots, p$). In other words, we allow the patterns to be alike —but not necessarily exactly equal— to one of the p representative patterns $\boldsymbol{\xi}^\mu$: we allow for what is called, technically, 'noise'.

In a preceding article, we studied patterns $\boldsymbol{x}$, belonging to sets of patterns Ω^μ clustered around typical patterns $\boldsymbol{\xi}^\mu$, in the context of basins of attraction [14]. In other words, we introduced Ω^μ as a means to construe, by hand, basins of attraction. In the present article, we use Ω^μ to represent noisy patterns. Thus the sets Ω^μ appearing in the two articles have the same meaning, namely sets of patterns around a typical pattern, but the reasons for introducing them is different: in the preceding article there was a mathematical motivation, whereas in the current article it is motivated by the biological reality that patterns are never exactly equal.

The purpose of this article is to determine the values for the weights w_{ij}, in case of learning with noisy patterns. We start by simply conjecturing that for noisy patterns the old rule (2) can essentially be maintained: all what we do is replacing in (2) the ξ^μ's by $\boldsymbol{x}$'s. Hence, we take as learning rule

$$\Delta w_{ij}(\boldsymbol{x}, t_n) = \eta_i[\kappa - \gamma_i(\boldsymbol{x}, \boldsymbol{w}_i(t_n))](2x_i - 1)x_j \qquad (j \in V_i). \qquad (6)$$

The learning rate η_i figuring in this expression will be discussed in section 3.4; the margin parameter κ has been discussed in a preceding article [14]. We will prove that this learning rule leads, on the average, to suitable values for the weights. From this we conclude that the energy saving learning rule is suitable also for learning the right patterns $\boldsymbol{\xi}^\mu$, on the basis of wrong (i.e. perturbed) input patterns $\boldsymbol{x}$ of Ω^μ.

2.2. The Discrete Time Master Equation

When learning takes place in a biological neural network with a learning rule of the type (6), we assume that the changes of the weights at time t_{n+1} depend only on the values of the weights $\boldsymbol{w}_i$ at time t_n and the variable $\boldsymbol{x} = (x_1, \ldots, x_n)$ which is randomly drawn out of the collection $\Omega = \cup_\mu \Omega^\mu$, the union of disjunct sets Ω^μ of patterns $\boldsymbol{x}$ centered around typical patterns $\boldsymbol{\xi}^\mu$, at time t_n. Consequently, learning with a learning rule like (6) is a Markovian process, and the weights w_{ij} are stochastic variables. Thus we have for the new weights w_{ij}':

$$w_{ij}' = \begin{cases} w_{ij} + \Delta w_{ij}(\boldsymbol{x}, \boldsymbol{w}_i) & (j \in V_i) \\ w_{ij} & (j \in V_i^c) \end{cases} \qquad (7)$$

where the $\Delta w_{ij}(\boldsymbol{x}, \boldsymbol{w}_i)$ are the increments given by the learning rule (6),

$$\Delta w_{ij}(\boldsymbol{x}, \boldsymbol{w}_i) = \eta_i[\kappa - \gamma_i(\boldsymbol{x}, \boldsymbol{w}_i)](2x_i - 1)x_j \qquad (j \in V_i). \tag{8}$$

Let $T_{ij}(w'_{ij}|w_{ij})$ be the probability density that a transition takes place from the value w_{ij} to the value w'_{ij}. Then we have

$$T_{ij}(w'_{ij}|w_{ij}) = \sum_{\boldsymbol{x} \in \Omega} p(\boldsymbol{x})\delta(w'_{ij} - w_{ij} - \Delta w_{ij}(\boldsymbol{x}, \boldsymbol{w}_i)) \tag{9}$$

where $p(\boldsymbol{x})$ is the probability to draw $\boldsymbol{x}$ from Ω. The δ-function guarantees that only transitions take place which obey the learning rule (8). Using a probability $p(\boldsymbol{x})$ normalized to unity, i.e.,

$$\sum_{\boldsymbol{x} \in \Omega} p(\boldsymbol{x}) = 1 \tag{10}$$

we find from (9) the following total transition probability to a state w_{ij}:

$$\int dw'_{ij}\, T_{ij}(w'_{ij}|w_{ij}) = 1\,. \tag{11}$$

Let $P_{ij}(w_{ij}, t_{n+1})$ be the probability of occurrence of the variable w_{ij} at t_{n+1}. Then, the probability P_{ij} and the transition probability T_{ij} are related according to

$$P_{ij}(w_{ij}, t_{n+1}) = \int dw'_{ij}\, T_{ij}(w_{ij}|w'_{ij})P_{ij}(w'_{ij}, t_n) \qquad (i = 1, \ldots, N; j \in V_i). \tag{12}$$

Let, moreover, the probability P_{ij} be normalized according to

$$\int dw_{ij}\, P_{ij}(w_{ij}, t_n) = 1\,. \tag{13}$$

From (11) and (12) it follows that

$$P_{ij}(w_{ij}, t_{n+1}) - P_{ij}(w_{ij}, t_n) = \int dw'_{ij}\, [T_{ij}(w_{ij}|w'_{ij})P_{ij}(w'_{ij}, t_n) \\ - T_{ij}(w'_{ij}|w_{ij})P_{ij}(w_{ij}, t_n)] \qquad (i = 1, \ldots, N; j \in V_i) \tag{14}$$

which is the Discrete Time Master Equation for the weights w_{ij}.

Next, let us consider the average of the weights at t_n:

$$\langle w_{ij} \rangle_{t_n} := \int dw_{ij}\, P_{ij}(w_{ij}, t_n)w_{ij} \qquad (j \in V_i) \tag{15}$$

or, using the normalization (13),

$$\langle w_{ij} \rangle_{t_n} = \prod_{k=1}^{N} \int dw_{ik}\, P_{ik}(w_{ik}, t_n)w_{ij} \qquad (j \in V_i)\,. \tag{16}$$

Using the Master Equation (14), we obtain for the change of the weights

$$\langle w_{ij} \rangle_{t_{n+1}} - \langle w_{ij} \rangle_{t_n} = \prod_{k=1}^{N} \int dw_{ik}dw'_{ik}\, [T_{ik}(w_{ik}|w'_{ik})P_{ik}(w'_{ik}, t_n) \\ - T_{ik}(w'_{ik}|w_{ik})P_{ik}(w_{ik}, t_n)]w_{ij}\,. \tag{17}$$

Interchanging the primed and unprimed variables in the first term we find

$$\langle w_{ij} \rangle_{t_{n+1}} - \langle w_{ij} \rangle_{t_n} = \prod_{k=1}^{N} \int dw_{ik}dw'_{ik}\, (w'_{ij} - w_{ij})T_{ik}(w'_{ik}|w_{ik})P_{ik}(w_{ik}, t_n) \tag{18}$$

or, with (9),

$$\langle w_{ij}\rangle_{t_{n+1}} - \langle w_{ij}\rangle_{t_n} = \sum_{\boldsymbol{x}\in\Omega} p(\boldsymbol{x}) \prod_{k=1}^{N} \int dw_{ik}\, \Delta w_{ij}(\boldsymbol{x},\boldsymbol{w}_i) P_{ik}(w_{ik},t_n) \qquad (j\in V_i)\,. \quad (19)$$

Inserting (8), and using (13) and (15) we find a first order difference equation for the variable $\langle w_{ij}\rangle_{t_n}$

$$\langle w_{ij}\rangle_{t_{n+1}} - \langle w_{ij}\rangle_{t_n} = \sum_{\boldsymbol{x}\in\Omega} p(\boldsymbol{x})\eta_i[\kappa - \gamma_i(\boldsymbol{x},\langle\boldsymbol{w}_i\rangle_{t_n})](2x_i-1)x_j \qquad (j\in V_i) \quad (20)$$

which can be solved once more is known about the probability $p(\boldsymbol{x})$.

Let $p^\mu(\boldsymbol{x})$ be the probability to draw $\boldsymbol{x}$ from the disjunct collections Ω^μ, normalized according to

$$\sum_{\boldsymbol{x}\in\Omega^\mu} p^\mu(\boldsymbol{x}) = 1\,. \quad (21)$$

Then the probability to draw $\boldsymbol{x}$ from $\Omega = \cup_\mu\Omega^\mu$ is

$$p(\boldsymbol{x}) = \frac{1}{p}\sum_{\mu=1}^{p} p^\mu(\boldsymbol{x}) \quad (22)$$

in agreement with the normalization (10), as may be verified with (21) and (22).

Let $p_i^\mu(x_i)$ be the probability that in the collection Ω^μ the i-th component of $\boldsymbol{x}$ has the value x_i. Then the probability $p_i^\mu(x_i)$ to draw from the collection Ω^μ the vector $\boldsymbol{x}$ is given by

$$p^\mu(\boldsymbol{x}) = \prod_{i=1}^{N} p_i^\mu(x_i)\,. \quad (23)$$

By choosing the normalization according to

$$\sum_{x_i=0,1} p_i^\mu(x_i) = 1 \quad (24)$$

we find that (21) is satisfied.

We now introduce the average with respect to the set Ω^μ:

$$\bar{x}_i^\mu := \sum_{\boldsymbol{x}\in\Omega^\mu} p^\mu(\boldsymbol{x})x_i\,. \quad (25)$$

In view of (22), (23), (24) and (25) we have

$$\sum_{\boldsymbol{x}\in\Omega} p(\boldsymbol{x})x_i = \frac{1}{p}\sum_{\mu=1}^{p} \bar{x}_i^\mu \quad (26)$$

$$\sum_{\boldsymbol{x}\in\Omega} p(\boldsymbol{x})x_i x_j = \frac{1}{p}\sum_{\mu=1}^{p} \bar{x}_i^\mu \bar{x}_j^\mu\,. \quad (27)$$

With these relations we may rewrite the difference equation for the average weights, equation (20), in the form

$$\langle w_{ij}\rangle_{t_{n+1}} - \langle w_{ij}\rangle_{t_n} = \frac{1}{p}\sum_{\mu=1}^{p} \eta_i[\kappa - \gamma_i(\bar{\boldsymbol{x}}^\mu,\langle\boldsymbol{w}_i\rangle_{t_n})](2\bar{x}_i^\mu-1)\bar{x}_j^\mu \qquad (j\in V_i) \quad (28)$$

This equation will be solved, in the next section, in the limit of large n. Note that the γ_i's now contain the averages $\bar{\boldsymbol{x}}^\mu$.

2.3. The Gauss-Seidel solution

The equation (28) for the average value of the weights w_{ij} at time t_n, can be solved, for $n \to \infty$, in a way that closely parallels the method of Diederich and Opper [15]. First, using (28) recursively, we arrive at

$$\langle w_{ij}\rangle_{t_{n+1}} = \langle w_{ij}\rangle_{t_0} + N^{-1}\sum_{\mu=1}^{p} F_i^{\mu}(t_n)\bar{x}_j^{\mu} \qquad (i = 1,\ldots,N; j \in V_i) \tag{29}$$

where

$$F_i^{\mu}(t_n) = \frac{\eta_i}{\alpha}\sum_{m=0}^{n}[\kappa - \gamma_i(\bar{\boldsymbol{x}}^{\mu},\langle\boldsymbol{w}_i\rangle_{t_m})](2\bar{x}_i^{\mu}-1) \tag{30}$$

with $\alpha = p/N$. From (30) it follows that

$$F_i^{\mu}(t_n) - F_i^{\mu}(t_{n-1}) = \frac{\eta_i}{\alpha}[\kappa - \gamma_i(\bar{\boldsymbol{x}}^{\mu},\langle\boldsymbol{w}_i\rangle_{t_n})](2\bar{x}_i^{\mu}-1) \tag{31}$$

or, using (3),

$$F_i^{\mu}(t_n) - F_i^{\mu}(t_{n-1}) = \frac{\eta_i}{\alpha}[\kappa(2\bar{x}_i^{\mu}-1) - (\sum_{k\in V_i}\langle w_{ik}\rangle_{t_n}\bar{x}_k^{\mu} + \sum_{k\in V_i^c}\langle w_{ik}\rangle_{t_0}\bar{x}_k^{\mu} - \theta_i)]. \tag{32}$$

Eliminating $\langle w_{ik}\rangle_{t_n}$ from (32) via (29) yields

$$\begin{aligned}\frac{\alpha}{\eta_i}(F_i^{\mu}(t_n) - F_i^{\mu}(t_{n-1})) =& [\kappa(2\bar{x}_i^{\mu}-1) - (\sum_{k=1}^{N}\langle w_{ik}\rangle_{t_0}\bar{x}_k^{\mu} - \theta_i)] \\ &- N^{-1}\sum_{k\in V_i}\sum_{\nu=1}^{p}F_i^{\nu}(t_{n-1})\bar{x}_k^{\nu}\bar{x}_k^{\mu}\end{aligned}. \tag{33}$$

To solve this set of linear equations, we shall rewrite them in matrix notation. First of all, let us introduce a $p \times p$ matrix, $\bar{C}_i$, with matrix-elements given by

$$\bar{C}_i^{\mu\nu} := N^{-1}\sum_{k\in V_i}\bar{x}_k^{\mu}\bar{x}_k^{\nu}. \tag{34}$$

We will refer to this matrix as the 'correlation matrix of averages'. The connection with a usual correlation matrix becomes more apparent in case $\bar{x}_k^{\mu}$ can be replaced by ξ_k^{μ}. Then the 'correlation matrix of averages' is identical to the 'reduced correlation matrix' [8]. We also introduce a $p \times p$ diagonal matrix H_i with diagonal elements given by $H_i^{\mu\mu} = \alpha/\eta_i$. Finally we shall denote a $p \times p$ unit matrix as I. Apart from the above mentioned matrices, we introduce the vectors $\mathbf{F}_i(t_n) := (F_i^1(t_n),\ldots,F_i^p(t_n))$ and $\mathbf{G}_i := (G_i^1,\ldots,G_i^p)$ with components $G_i^{\mu} = [\kappa(2\bar{x}_i^{\mu}-1) - (\sum_{k=1}^{N}\langle w_{ik}\rangle_{t_0}\bar{x}_k^{\mu} - \theta_i)]$. With these notations and abbreviations (33) can be recast in the simple form

$$H_i \cdot \mathbf{F}_i(t_n) = (H_i - \bar{C}_i)\cdot\mathbf{F}_i(t_{n-1}) + \mathbf{G}_i \tag{35}$$

Solving this equation iteratively for $\mathbf{F}_i(t_n)$ we obtain

$$\begin{aligned}\mathbf{F}_i(t_n) =& [H_i^{-1}\cdot(H_i-\bar{C}_i)]^n\mathbf{F}_i(t_0) + H_i^{-1}[I + H_i^{-1}\cdot(H_i-\bar{C}_i) \\ &+ \ldots + [H_i^{-1}\cdot(H_i-\bar{C}_i)]^{n-1}]\cdot\mathbf{G}_i\end{aligned} \tag{36}$$

The matrix $\bar{C}_i$, as defined in (34), is easily seen to be positive definite and symmetric. It then can be shown that the matrix $H_i^{-1}\cdot(H_i - \bar{C}_i)$ has eigenvalues smaller than one [16]. As a consequence, we have

$$\lim_{n\to\infty}[H_i^{-1}\cdot(H_i-\bar{C}_i)]^n = 0. \tag{37}$$

This implies that, in the limit $n \to \infty$, (36) converges to

$$
\begin{aligned}
\mathbf{F}_i(t_\infty) &= H_i^{-1}[I - H_i^{-1} \cdot (H_i - \bar{C}_i)]^{-1} \cdot \mathbf{G}_i \\
&= \bar{C}_i^{-1} \cdot \mathbf{G}_i \,.
\end{aligned}
\tag{38}
$$

Substitution of (38), in index-notation, into (29) yields the following result for the average weights $\langle w_{ij} \rangle_{t_\infty}$ $(j \in V_i)$ after a learning process with noisy patterns

$$
\langle w_{ij} \rangle_{t_\infty} =
\begin{cases}
\langle w_{ij} \rangle_{t_0} + v_{ij} & (j \in V_i) \\
\langle w_{ij} \rangle_{t_0} & (j \in V_i^c)
\end{cases}
\tag{39}
$$

where

$$
v_{ij} = N^{-1} \sum_{\mu,\nu=1}^{p} [\kappa - \gamma_i(\bar{x}^\mu, \langle \boldsymbol{w}_i \rangle_{t_0})](2\bar{x}_i^\mu - 1)(\bar{C}_i^{-1})^{\mu\nu}\bar{x}_j^\nu \,.
\tag{40}
$$

Hence, although the weights $w_{ij}(t_n)$ themselves do not converge, since the increments $\Delta w_{ij}(t_n)$, given by eq. (6), do not tend to zero for n going to infinity, the average $\langle w_{ij}(t_n) \rangle$ does. We thus find that the actual, biological weights $w_{ij}(t_n)$ of the neural net fluctuate around the average value $\langle w_{ij} \rangle_{t_\infty}$ (see figure 2). The expression (39)–(40) for the average weights of a network trained with noisy patterns constitutes the main analytical result of this article.

In the following section we carry out a numerical analysis on the process of learning patterns that are perturbed. We will use a particular expression for the probability distribution $p^\mu(\boldsymbol{x})$, namely, eq. (43), which is the same as the one used in a preceding article [14], in order to compare the biological result (39)–(40) for the weights and the result for the weights obtained in case of a mathematical approach aiming at creating fixed points with prescribed basins of attraction.

3. Numerical analysis

On the basis of the result found above, in particular equations (39)–(40), we expect that the energy saving learning rule (6) applied to a set of noisy input patterns $\boldsymbol{x} \in \Omega^\mu$ will lead to satisfactory results. By satisfactory we mean that the system can recognize patterns belonging to the clusters of patterns Ω^μ. Still more in detail we mean that, after a certain number of learning steps, each cluster Ω^μ has a fixed point, $\boldsymbol{y}^\mu$ say, whereas the other patterns of Ω^μ belong to the basin of attraction of this fixed point $\boldsymbol{y}^\mu$. It should be noted that when a new pattern $\boldsymbol{z}^\mu \in \Omega^\mu$ is learned, the old fixed point $\boldsymbol{y}^\mu \in \Omega$ can be replaced by a new fixed point $\boldsymbol{z}^\mu$. This is a direct consequence of the learning rule (6) with a pre-factor given by (48), which is such that the last learned pattern becomes automatically, and in one learning step only, a fixed point of the dynamics, see the preceding article [8]. In this preceding article, we considered learning of p patterns $\boldsymbol{\xi}^\mu$ $(\mu = 1, \ldots, p)$. In the language of the present article, we can say that we studied, in the preceding article, the learning of 'clusters' Ω^μ $(\mu = 1, \ldots, p)$, each consisting of one single pattern, $\boldsymbol{y}^\mu = \boldsymbol{\xi}^\mu$. Since there was only one pattern per cluster Ω^μ, the fixed point remained the same during all of the learning process. In this article, the situation is a little bit different: the fixed point of a cluster Ω^μ is 'roaming' over Ω^μ, i.e., the fixed point is no longer fixed during all of the learning process.

3.1. A measure for the performance of a neural net

A criterium for the way in which a neural net functions may be based on the stability coefficients γ_i. The more of them are positive, for given sets of typical patterns $\boldsymbol{\xi}^\mu$, the better the net fulfils its task of storing and recollecting patterns [8]. Inserting the final expression for the weights, eqs. (39)–(40), into the definition of the stability coefficients γ_i, eq. (3), it follows that at t_∞ they are given by

$$\gamma_i(\bar{\boldsymbol{x}}^\mu, \langle \boldsymbol{w}_i \rangle_{t_\infty}) = \kappa \qquad (i = 1, \ldots, N; \mu = 1, \ldots, p). \tag{41}$$

In view of (27), the γ-function of the average $\bar{\boldsymbol{x}}^\mu$ can be replaced by the average of the γ-function of $\boldsymbol{x}$ itself

$$\bar{\gamma}_i^\mu(\boldsymbol{x}, \langle \boldsymbol{w}_i \rangle_{t_\infty}) = \kappa \qquad (i = 1, \ldots, N; \mu = 1, \ldots, p). \tag{42}$$

Since the margin parameter κ is positive, the latter equation implies that, on the average, the γ_i are positive. We now recall that for a perfectly functioning network *all* γ_i should be positive. Therefore, by calculating the fraction of γ_i's that are positive in various cases, we can judge the quality of a neural network.

3.2. A useful probability distribution

We now address the question whether there exists, after a certain number of learning steps, a (roaming) fixed point for each cluster. An alternative way of putting this question is to ask whether there exist $\boldsymbol{z}^1, \ldots, \boldsymbol{z}^p$ such that the $\gamma_i(\boldsymbol{z}^\mu, \boldsymbol{w}_i(t))$ are all positive at a certain time t. We will investigate this question numerically. To that end, we choose a particular form for the clusters Ω^μ by specifying the choice of the probability distribution $p^\mu(\boldsymbol{x})$, equations (22)–(23). We take for its i-th factor

$$p_i^\mu(x_i) = (1 - b)\,\delta_{x_i, \xi_i^\mu} + b\,\delta_{x_i, 1 - \xi_i^\mu} \tag{43}$$

where b is a parameter between 0 and 1, which we will refer to as the 'noise-parameter'. If $b = 0$ (no noise), only the patterns $\boldsymbol{x} = \boldsymbol{\xi}^\mu$ have a non-zero probability of occurrence. For values of b close to zero any vector $\boldsymbol{x}$ has a non-zero probability of occurrence, but only vectors $\boldsymbol{x}$ close to one of the $\boldsymbol{\xi}^\mu$ have a probability of occurrence comparable to the probability of occurrence of a typical pattern. The particular choice (43) for $p^\mu(\boldsymbol{x})$ enables us to construct the collection of vectors $\boldsymbol{x}$ to be used as learning input vectors in our numerical calculation. Since in the derivation of the Master Equation the clusters Ω^μ have been chosen disjunct, a vector $\boldsymbol{x}$ cannot belong to more than one cluster. According to the probability distribution (43), however, a vector $\boldsymbol{x}$ which belongs to a certain cluster Ω^μ, has a —very small, but— non-zero probability to belong to any other cluster. This implies that (43) is not exact but only a —very good— approximation to the actual situation, for which these probabilities vanish exactly.

The expression (43) can be used to calculate the average $\bar{x}_i^\mu$. Inserting (22) with (23) and (43) into (25) we obtain

$$\bar{x}_i^\mu = \sum_{x_i = 0, 1} \left[(1 - b)\,\delta_{x_i, \xi_i^\mu} + b\,\delta_{x_i, 1 - \xi_i^\mu} \right] x_i$$
$$\times \prod_{k \neq i} \sum_{x_k = 0, 1} \left[(1 - b)\,\delta_{x_k, \xi_k^\mu} + b\,\delta_{x_k, 1 - \xi_k^\mu} \right] \tag{44}$$

or

$$\bar{x}_i^\mu = (1 - b)\xi_i^\mu + b(1 - \xi_i^\mu). \tag{45}$$

For $b = 0$, the case of patterns without noise, $\bar{x}_i^\mu$ reduces to ξ_i^μ. Using this fact in our present main analytical result, given by equations (39)–(40), one indeed recovers the old result (4) for the final values of the weights in case of noiseless input.

It is also instructive to compare the result for the weights $\langle w_{ij} \rangle_{t_\infty}$, equations (39), (40) with (45) and the result of the preceding article [14]. In the latter article, we calculated the weights of a recurrent neural net, denoted as $w_{ij}(t)$, in case fixed points and basins of attraction are taken explicitly into account, see equations (1)–(2) of [14]. The only difference is found in the factor $2\bar{x}_i^\mu - 1$, which, in our result based upon basins, reads $2\xi_i^\mu - 1$. In fact, we find for the difference of the preceding and present results:

$$w_{ij}(t) - \langle w_{ij} \rangle_{t_\infty} = 2\kappa N^{-1} \sum_{\mu,\nu=1}^{p} (\xi_i^\mu - \bar{x}_i^\mu)(\bar{C}_i^{-1})^{\mu\nu} \bar{x}_j^\nu \tag{46}$$

where we used the expressions (1)–(2) of [14] and (39)–(40), together with the definitions of γ in the two cases. Furthermore, we put $\langle w_{ij} \rangle_{t_0} = w_{ij}(t_0)$ for the initial values of the weights. Since

$$\xi_i^\mu - \bar{x}_i^\mu = b(2\xi_i^\mu - 1) \tag{47}$$

the difference (46) is of the order b, i.e., small compared to the weights themselves. Consequently, the biological system considered here is found to be able to realize the optimal values $w_{ij}(t)$ for the weights derived earlier [14], up to terms that are small compared to unity. This is intriguing, since, a priori, there is no reason to expect that a biological learning rule based on economy of energy to rebuild a synapse [8] will lead to values of the weights that are a good approximation to the values found from the requirement that there are fixed points with prescribed basins [14].

Due to the fact that the final results of [14] and this article for the weights are very similar, one may expect that natural, biologically learning via the learning rule (6) for noisy patterns $\boldsymbol{x} \in \Omega^\mu$ ($\mu = 1, \ldots, p$) will lead to larger basins of attraction than in case of learning of noiseless patterns.

3.3. Storage of noisy patterns

Having chosen the sets Ω^μ, via $p_i^\mu(x_i)$, eq. (43), we are able to simulate a learning process with noisy patterns. What we will do in the numerical study below is to pick an input vector $\boldsymbol{x}$ according to a probability distribution as given by equation (43). Next, we calculate the new synaptic weights $w_{ij}(t_{n+1})$, equation (1), using the energy saving learning rule (6). Finally, we calculate the pN stability coefficients $\gamma_i(\boldsymbol{z}^\mu, \boldsymbol{w}_i(t_{n+1}))$ ($i = 1, \ldots, N$), where $\boldsymbol{z}^\mu$ stands for the last learned pattern of Ω^μ ($\mu = 1, \ldots, p$), coefficients which we hope to be positive. The result is shown in figure 1.

The left and right columns in figure 1 correspond to two particular choices for the factor η_i occurring in the learning rule (6), namely the 'local' and 'global' learning rules, which will be described in section 3.4. Going downwards in one of these two columns, the number of learning steps rises. In each figure, we have put the number of gamma's as a function of its value. It is seen that after 300 learning steps almost all γ's are positive. Hence, at the 300-th step, most of the last learned patterns $\boldsymbol{z}^\mu$ are fixed points indeed. We note that it is instructive to compare the results of learning of patterns with and without noise: see figure 3 of [8]. There is almost no difference in case of the local learning rule, whereas noise seems to diminish a little bit the effectiveness of the (non-biological) global learning rule.
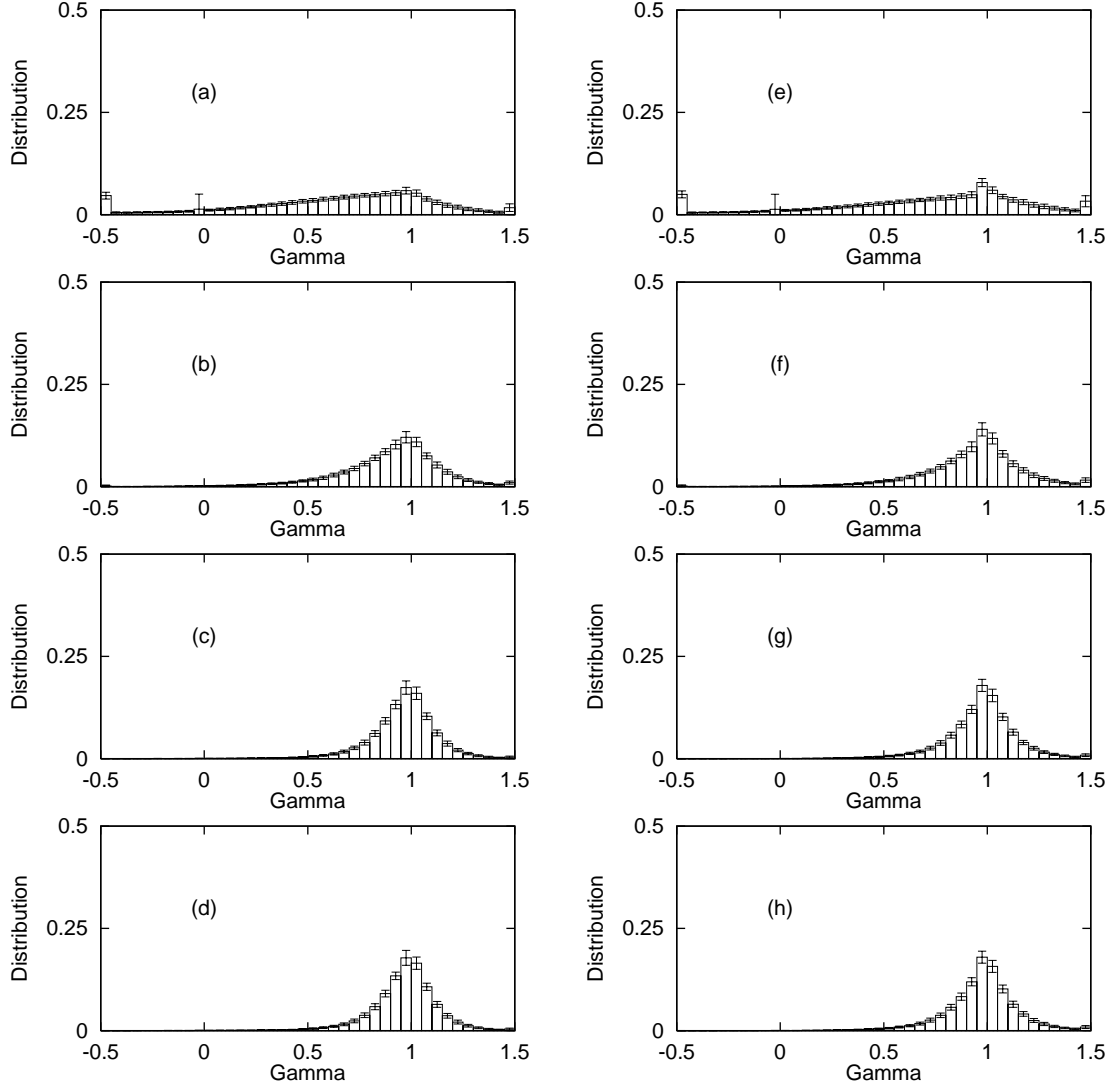
Figure 1. Noisy patterns. The performance of a neural network is measured by the stability coefficients γ_i related to neuron i, which should be positive for a properly functioning neural net. In the figure we have plotted averages of these γ's, for series of noisy input patterns, for the local learning rule (left column) and the global learning rule (right column). The number of learning steps increases from 32 (top) via 160 and 320 to 640 (bottom). The average is taken over 100 sets of $p = 32$ patterns for a neural network with $N = 128$ neurons. The noise parameter which yields the sets Ω^μ is taken to be $b = 0.01$. The calculations have been performed starting with a tabula rasa for the weights: $w_{ij}(t_0) = 0$, and for neurons with vanishing thresholds: $\theta_i = 0$. The dilution in the network is $d = 0.2$, the average activity of the net is $a = 0.2$. The normalization of the weights has been fixed such that $\kappa = 1$. Both in case of global and local learning, the proposed learning rule (6) leads to a very satisfactory result: almost all γ's become positive.
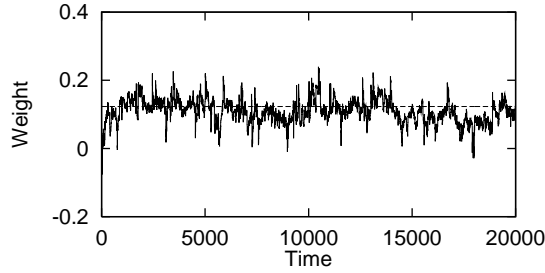
Figure 2. **Fluctuations of the weight of a connection around its average**. The value of an arbitrarily chosen synaptic connection w is followed during $20,000$ learning steps, for the local learning rule. A similar picture is obtained in case of the global learning rule. The average value $\langle w \rangle_{t_\infty}$ as predicted by equations (39)–(40) is depicted as a horizontal dashed line. It is observed that the value of the connection fluctuates around the dashed line. The same parameters are chosen as in figure 1, i.e., $N = 128$, $p = 32$, $a = 0.2$, $b = 0.01$, $d = 0.2$, $\kappa = 1$, $w_{ij}(t_0) = 0$ and $\theta_i = 0$ for all i and j.

Finally, to illustrate the fact that the weights of the synaptic connections fluctuate around the average value as given by the expression (39)–(40), we have plotted, in figure 2, the time-evolution of the weight of an arbitrarily chosen connection together with its average value.

3.4. The learning rate η_i

So far, we have been concerned with learning, and problems related to learning. In our study of the learning process, the weights w_{ij} of the synaptic connections changed according to the learning rule (6), in which a factor η_i occurred, the so-called learning rate, which, so far, was left unspecified. In this subsection, we focus the attention on this factor η_i. In a preceding article, we showed that in a process of ideal learning, i.e., such that the energy needed to change the synapses is minimal, the learning rate η_i was given by

$$\eta_i = 1/\sum_{k \in V_i} x_k \,. \tag{48}$$

Depending on neuron activity not restricted to two neurons only, this factor is non-local and therefore biologically unrealistic (see [8], section 6). In a biological context, it should be replaced by some local approximation, for instance, a constant like

$$\eta_i = 1/Na \qquad (i = 1, \ldots, N) \tag{49}$$

where a denotes what is called the mean-activity, i.e., it is the probability that an arbitrary neuron i is in the state $+1$. In figure 1, the left column of pictures corresponds to the biological, local learning rule, i.e., equation (6) with η_i given by (49), while the right column corresponds to the global learning rule, i.e., equation (6) with η_i given by (48).

3.5. Retrieval by a biological network with ever changing connections

Is there more one can say on the value for the learning rate η_i in case of a biological network than that it should be an approximation to the value (48), which guarantees that the process of learning takes place in an energetically most economical way? The

answer is affirmative in case one requires that the learning is good enough to store patterns, but is not that good that it stores each learned pattern in only one learning step, as is the case for the global learning rule, i.e., the learning rule with η_i given by equation (48). In other words, it will turn out in this section that it is of advantage to learn via a learning rule that is not able to always store a new pattern in only one learning step. The reason for this counter-intuitive requirement which we are going to impose, is a consequence of the fact that we demand that the network be able to retrieve patterns *and* change connections at the same time.

In most models of neural networks one distinguishes between a learning phase and a retrieval phase. In the learning phase the weights are changed according to some rule, in the retrieval phase the weights are kept fixed. In a biological neural network such a separation of phases does not occur. Weights do not stop changing in the retrieval phase when a stimulus is presented, and this is precisely what is happening when a neural network has to recognize a pattern. If the change due to the stimulus would be too close to the 'ideal' value (48), the network would change in such a way that every new pattern would immediately be learned, and, hence, be recognized. And this is not what should be the case: if every new pattern would be stored immediately, it could not easily be distinguished from a pattern that had been stored in the network a long time ago already. Therefore, we must require that η_i is sufficiently unequal to the value (48), which it has in case of the global learning rule. If we take η_i larger and larger with respect to the value (48), network changes will become too large for the network to function properly [1]. So we are left with the possibility that η_i has a value somewhere between zero and the value (48), which is large enough to store patterns, and small enough for the network to distinguish between new and formerly learned patterns.

The above qualitative statements should now be made quantitative. In figure 3 we consider the storage of one pattern. We have plotted the percentage of positive γ's as a function of the learning rate η. For η in the range $(3/N, 11/N)$ all, or almost all, γ's are positive after one learning step. For $\eta \approx 1/N$, only 80% of the γ's are positive after one learning step. We conclude from all this that the factor η figuring in the learning rule (6) should be of the order of $3/N$ or less. Such a value guarantees that a biological network, which is bound to change its connections also during retrieval, does not learn so fast that it recognizes patterns already after one learning step, as in the case of the global learning rule (48).

In order to compare what happens when an already learned pattern is presented to the network with what happens in the net when a totally new pattern is the input, it is useful to define the overlap function of the two input patterns. The 'overlap', $Q(\boldsymbol{x}, \boldsymbol{y})$, of two binary patterns $\boldsymbol{x}$ and $\boldsymbol{y}$ of N bits defined in the usual way, is given by

$$Q(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{N} \sum_{i=1}^{N} (2x_i - 1)(2y_i - 1). \tag{50}$$

If $x_i = y_i$ for all $(i = 1, \ldots, N)$, the overlap takes its maximal value $+1$; if $x_i = 1 - y_i$, for all i, the overlap takes its minimal value -1. In figure 4 we compare the functioning of a neural network that changes its connections during the process of 'recognition' of previously learned patterns (left column) and random, non-learned, patterns (right column), for values of η going down from $3/N$ (top) to $1/N$ (bottom). In the left column we have plotted, vertically, the overlap $Q(\boldsymbol{x}(t_n), \boldsymbol{z}^{\mu}(t_n))$ of an arbitrary learned pattern $\boldsymbol{x}(t_n) \in \Omega$ that is presented to the net, and the last learned fixed point $\boldsymbol{z}^{\mu}$.
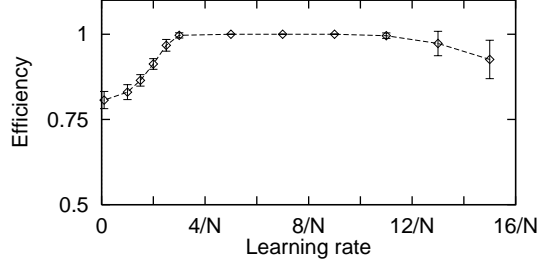
Figure 3. **Storage efficiency for local learning**. The neural net has been initialized by taking a tabula rasa for the weights, followed by some learning process. Then, the fraction of positive γ's, after one more learning step, has been plotted as a function of the learning rate η. For a biological network with ever changing connections η should not be too effective, i.e., not in the range $3/N$ to $11/N$. We took a neural net with $N = 512$ neurons, we used 20 initial learning steps. Furthermore, we took thresholds $\theta_i = 0$, dilution $d = 0.2$, activity $a = 0.2$ and noise level $b = 0.01$.

Both may change permanently in time because of the continuous updating of the weights. In the right column we have plotted, vertically, the overlap $Q(\boldsymbol{x}(t_n), \boldsymbol{x}(t_{n+1}))$ of an arbitrary pattern $\boldsymbol{x}(t_n)$, not previously presented to the network and the pattern $\boldsymbol{x}(t_{n+1})$ generated by the network one time step later. In both columns, time steps are plotted along the horizontal axis.

In our numerical study, the connections $w_{ij}(t_n)$ are changed according to the rule (1) combined with the learning rule (6); the patterns $\boldsymbol{x}(t_n)$ are updated using the usual dynamics [18, page 20]

$$x_i(t_{n+1}) = \Theta_{\mathrm{H}}(\sum_{j=1}^{N} w_{ij}(t_n)x_j(t_n) - \theta_i) \qquad (i = 1, \dots, N) \tag{51}$$

applied parallelly, i.e., at a time t_n, all neurons i update their states $x_i(t_n)$ simultaneously. The learned patterns $\boldsymbol{x}$ are chosen according to the probability $p^\mu(x)$ around $\boldsymbol{\xi}^\mu$, equation (43). The arbitrary, non-learned, patterns $\boldsymbol{x}$ are chosen randomly with mean activity $a = 0.2$.

In each of the four pictures of the left column learned patterns are presented to the network, and followed during ten learning steps. In the top left picture no recognition takes place, whereas for lower values of η the recognition capability of the network rises. In the bottom left picture recognition always takes place. In case recognition takes place for an $\boldsymbol{x} \in \Omega^\mu$, it is found that the fixed point $\boldsymbol{z}^\mu$ does not change in time. If, however, recognition does not occur, the fixed point $\boldsymbol{z}^\mu$ was found to change in time. Observe that in some cases the learned patterns seem to evolve to a two-state attractor, in contrast to what one might expect. In fact, we showed in article [8] that when a new pattern is learned with the global form of learning rule (2), one arrives at a one-state attractor (fixed point) after one learning step: see the first new paragraph of [8] under equation (43). Hence, we may expect that in the left column, where we use a local learning rule that approximates the global one, one-state attractors would occur only. The occurence of two-state attractors can only be a consequence of the fact that, in contrast to the treatment of [8], the weights always change in time and/or the approximation of the global learning rule by a local learning rule.

In case random patterns are the input to the network, there is in general no evolution of the network to one of the fixed points $\boldsymbol{z}^\mu$: a numerical study of the
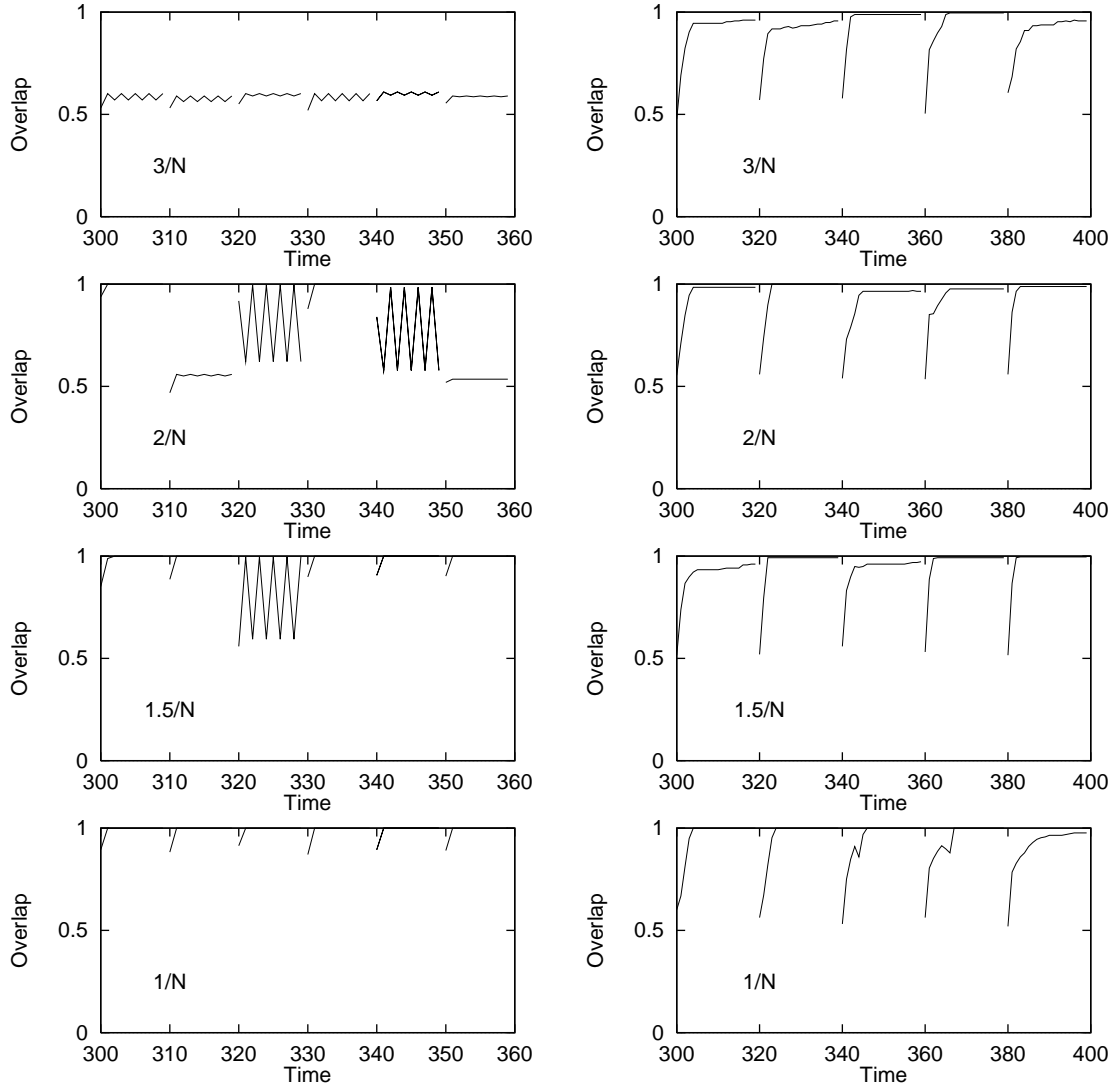
Figure 4. **Retrieval of learned and random patterns**. In each of the pictures at the left, a learned pattern is presented six times to a network, for $\eta = 3/N$, $2/N$, $1.5/N$ and $1/N$ (top to bottom). The recognition is preceded by a learning stage which took 300 learning steps. A pattern is recognized when the overlap is 1. In each of the pictures at the right, a random picture is presented five times to the network, after a learning period of the same number of 300 time-steps. Again η varied from $3/N$ to $1/N$ (top to bottom). A pattern is seen to evolve almost always to some other, stable, pattern, for all values of η, since the overlap with the preceding pattern almost always tends to 1. The network considered had the following network properties. Number of neurons $N = 512$, number of patterns $p = 16$, mean activity $a = 0.2$, dilution $d = 0.2$, noise parameter $b = 0.05$. The parameter η occurs in the learning rule (6) used to perform the updating of the weights of the connections.

overlap of a random pattern and any of the z^μ turned out to yield an overlap which was always less than 1. What we have pictured in the right column of figure 4 is evolution of a random pattern during 20 time steps. For all values of η a random pattern evolves to a pattern that remains stable or almost stable under the network dynamics. In related cases (see, e.g., [17] and [18] section 4.1) these 'spurious states' are found to vanish when the dynamics of the network is taken to be stochastic rather than deterministic.

Not only in case of learned patterns but also in case of random patterns fixed points z^μ have been found which change in time. This is due to the fact that the weights change continuous in time or due to the approximation of the global learning rule by a local learning rule. We do not pursue this and other points related to figure 4 any further.

4. Conclusions

The basis of this article is the learning rule for noisy patterns, equation (6). We found, by a numerical study of this learning rule, that storage of noisy patterns leads to fixed points that move around in collections Ω^μ that are representative for the noisy patterns. An analytical study of the same learning rule reveals that the weights found via this rule fluctuate, as long as learning or retrieval of patterns takes place, around certain average values, for which the explicit expression given by (39)–(40) could be derived.

In the limit of vanishing noise in the input, we recover the expressions for the weights obtained earlier [8] on the basis of a totally different approach, namely, economy of energy in case of synaptic change. This is satisfactory, because it yields an independent check of their correctness.

A comparison with other results obtained earlier [14], in which we determined the optimal weights for a neural net with prescribed basins of attraction, shows that the biological updating rule of the present article, eq. (6), realizes the latter results via eqs. (39)–(40) up to terms of the order of the noise parameter b, which are small compared to one.

References

[1] Lattanzi G, Nardulli G, Pasquariello G and Stramaglia S 1997 *Phys Rev E* **56** 4567
[2] Derrida B, Gardner E and Zippelius A 1987 *Europhys Lett* **4** 167
[3] Amit D J, Evans M R, Horner H and Wong K Y M 1990 *J Phys A: Math Gen* **23** 3361
[4] Fusi S and Amit D J 1992 *Int J of Neural Systems* **3** 3
[5] Caroppo D and Stramaglia S 1998 *Phys Lett A* **246** 55
[6] Amit D J and Brunel N 1995 *Network: Computation in Neural Systems* **6** 359
[7] Penney R W, Coolen A C C and Sherrington D 1993 *J Phys A: Math Gen* **26** 3681; Coolen A C C, Penney R W and Sherrington D 1993 *Phys Rev B* **48** 16117; Penney R W and Sherrington D 1994 *J Phys A: Math Gen* **27** 4027
[8] Heerema M and van Leeuwen W A 1999 *J Phys A: Math Gen* **32** 263
[9] van Kampen N G 1985 *Stochastic processes in physics and chemistry* (Amsterdam: North-Holland)
[10] Leen T K, Schottky B and Saad D 1999 *Phys Rev E* **59** 985
[11] Heskes T M and Kappen B 1991 *Phys Rev A* **44** 2718
[12] Radons G 1993 *J Phys A: Math Gen* **26** 3455
[13] Heskes T 1994 *J Phys A: Math Gen* **27** 5145
      , for the so-called stability coefficients.
[14] Heerema M and van Leeuwen W A submitted to *J Phys A: Math Gen*
[15] Diederich S and Opper M 1987 *Phys Rev Lett* **58** 949

[16] Carnahan B, Luther H A and Wilkes J O 1969 *Applied Numerical Methods* (New York: Wiley) 299

[17] Amit D J, Gutfreund H and Sompolinsky H 1985 *Phys Rev A* **32** 1007

[18] Amit D J 1989 *Modeling Brain Function* (Cambridge: Cambridge University Press)